C-based Synthesis of Area-Efficient Accelerators for OpenMP Worksharing Loops

Lukas Sommer, Julian Oppermann, Andreas Koch

```
#pragma omp parallel for shared(A,B,C) private(j,k) \
        num_threads(4) schedule(static)
    for(i=0; i<N; i++){
        for(j=0; j<N; j++){
            for(k=0; k<N; k++)
                C[i][j] = C[i][j] + A[i][k] * B[k][j];
        }
    }
```

Embedded Systems & Applications

TECHNISCHE UNIVERSITÄT DARMSTADT

Motivation

Exploitation of Thread-Level Parallelism (TLP) in High-Level Synthesis:

Motivation

Exploitation of Thread-Level Parallelism (TLP) in High-Level Synthesis:

- HPC usage scenarios demand higher throughput

Motivation

Exploitation of Thread-Level Parallelism (TLP) in High-Level Synthesis:

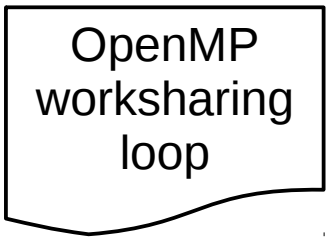
- HPC usage scenarios demand higher throughput
- FPGA sizes continue growing
  - ➔ Allows for duplicated datapaths

Motivation

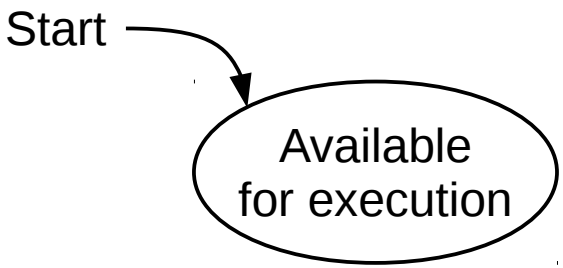
Exploitation of Thread-Level Parallelism (TLP) in High-Level Synthesis:

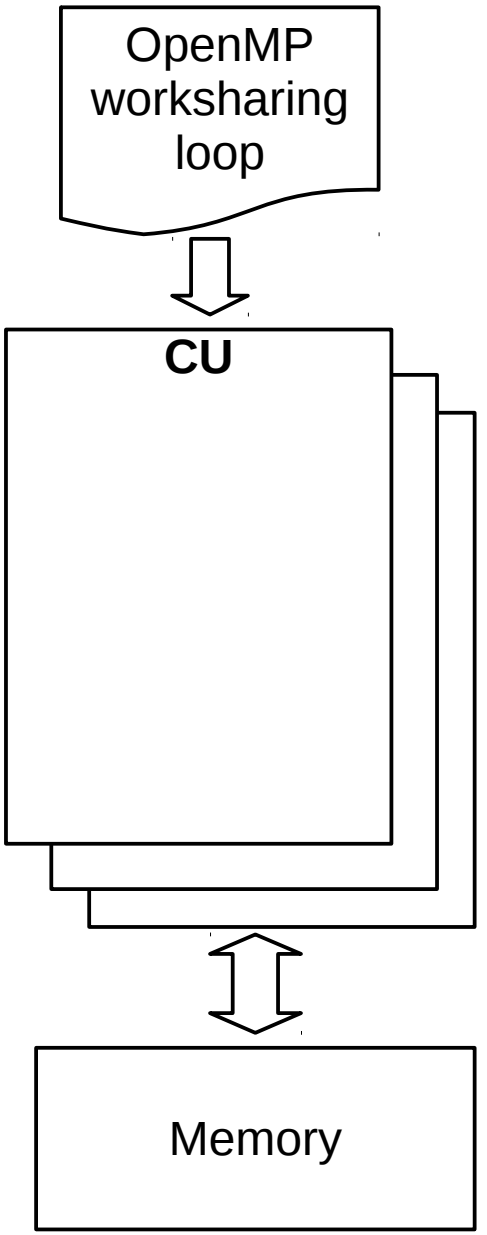
- HPC usage scenarios demand higher throughput
- FPGA sizes continue growing
  - ➔ Allows for duplicated datapaths
- Low utilization in HLS-generated compute units (CU) observed
  - ➔ Improve with dynamic thread interleaving

Nymble-OMP

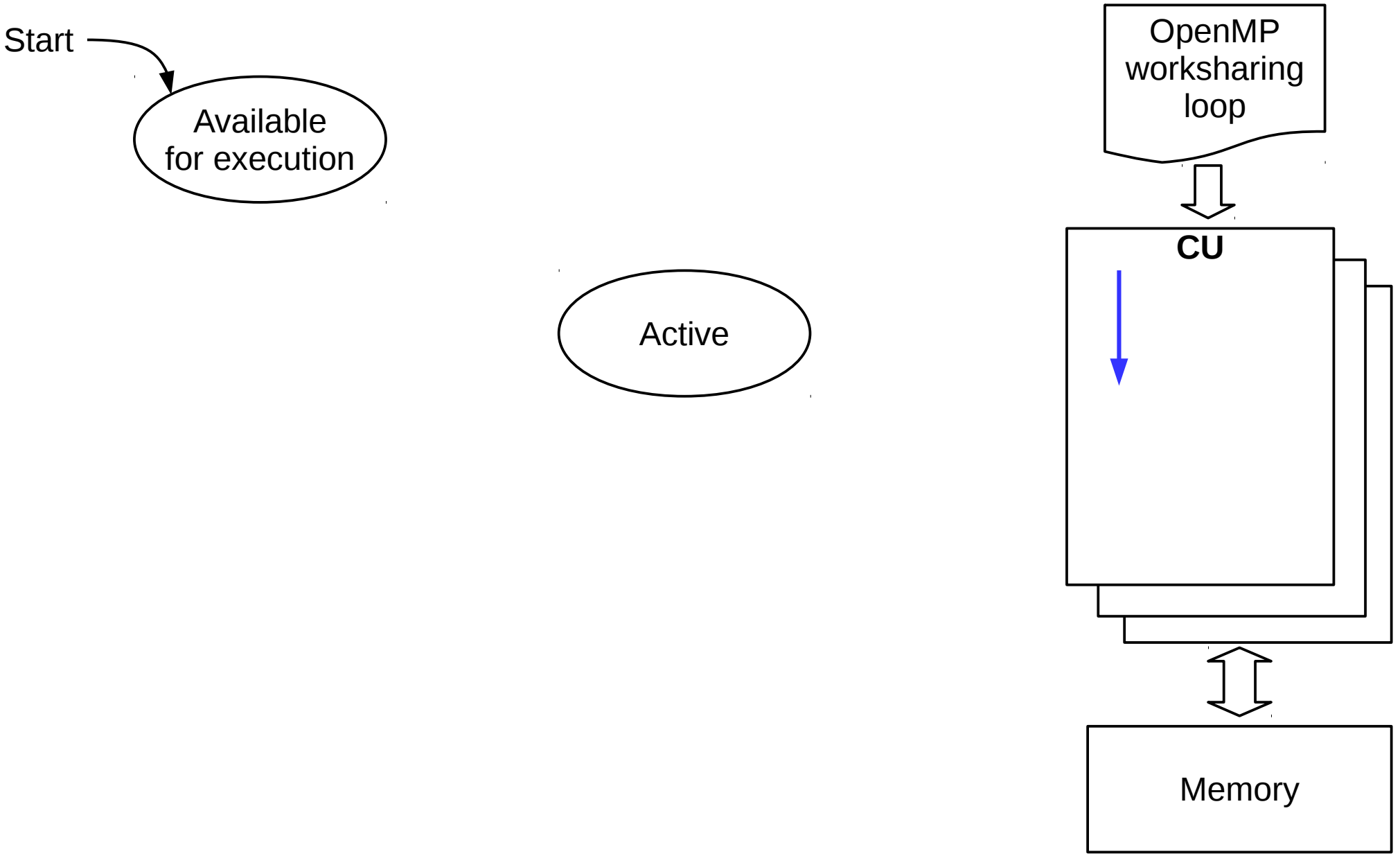


Nymble-OMP

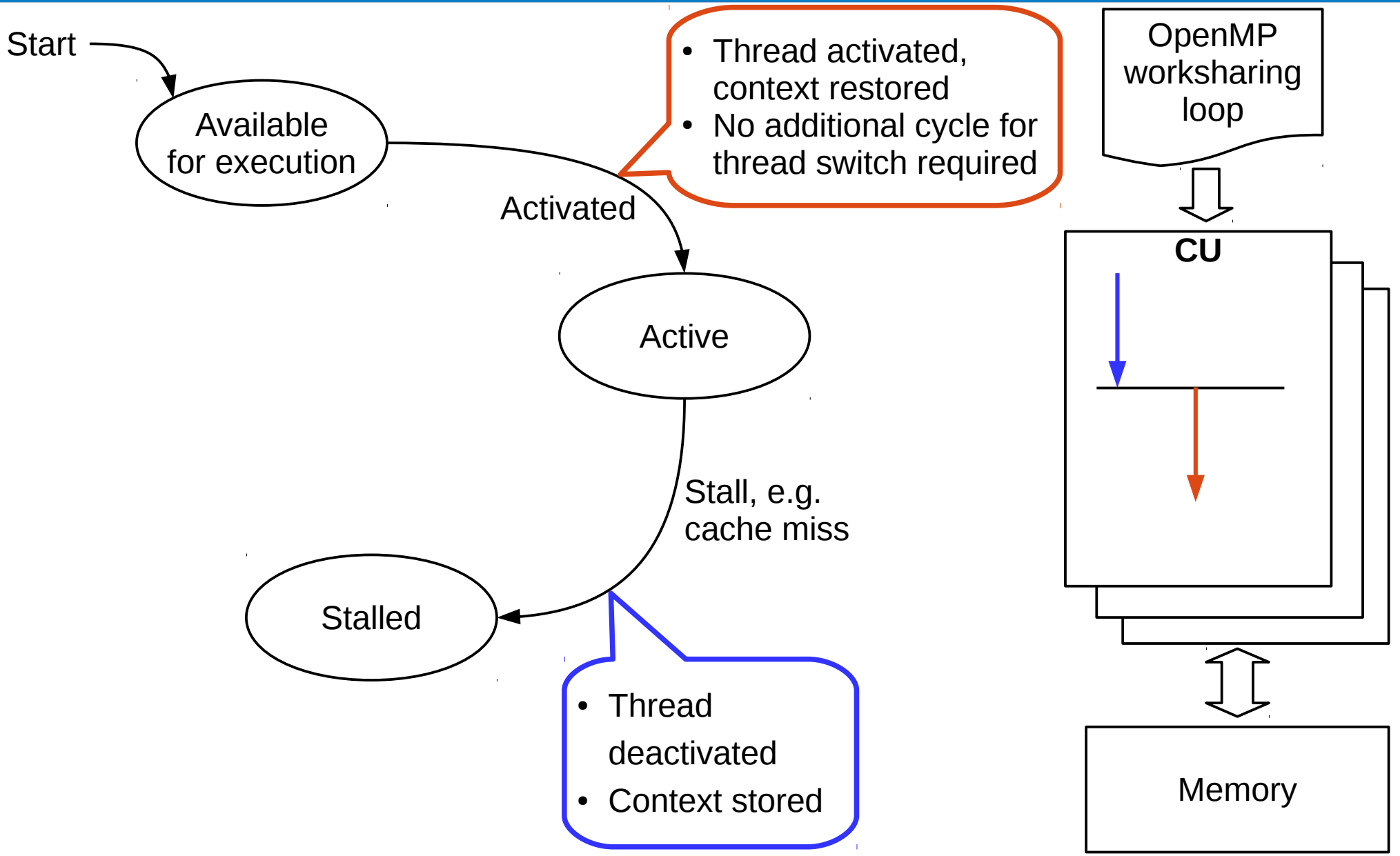

Start
Available
for execution


OpenMP
worksharing
loop
CU
Memory

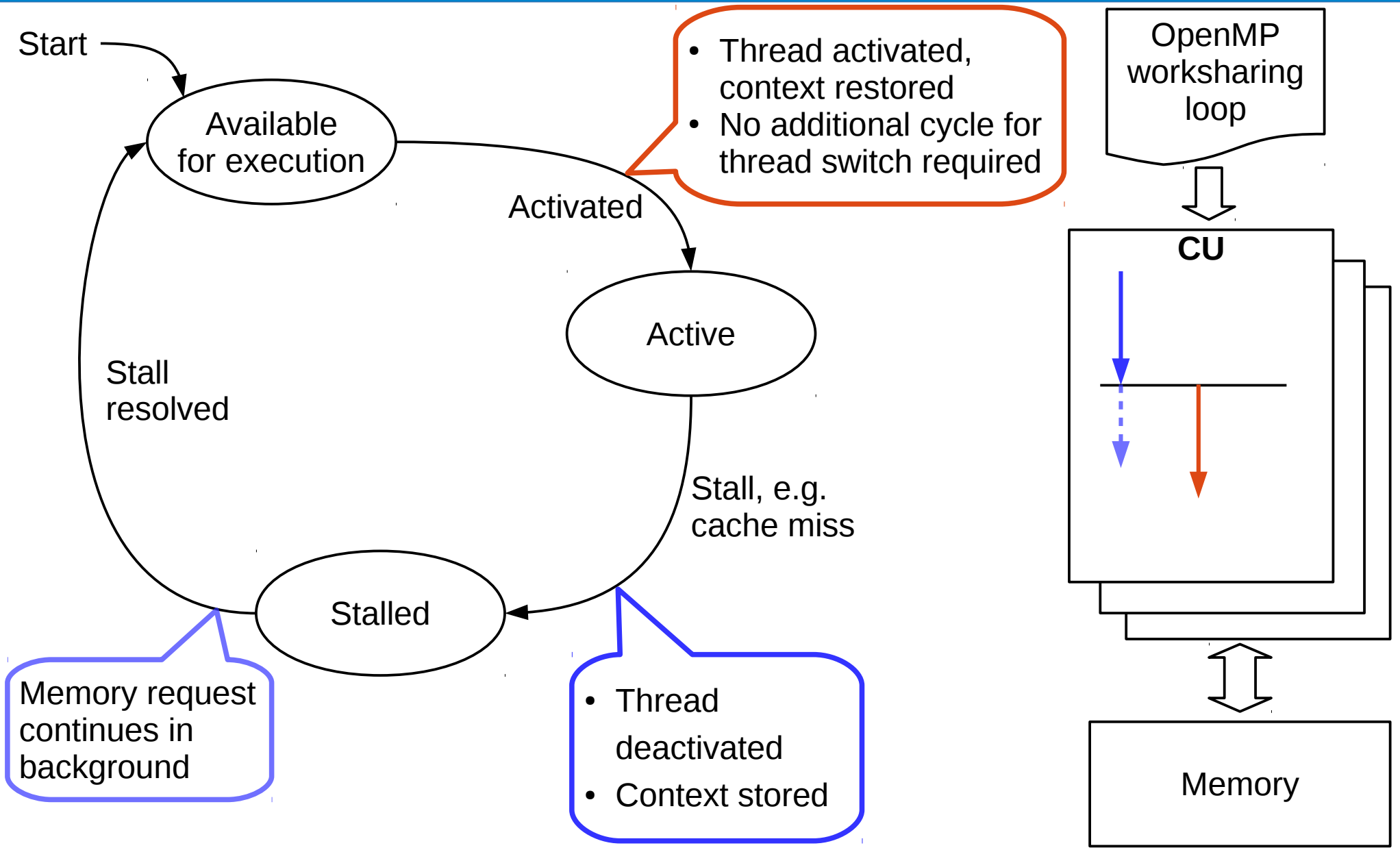
Nymble-OMP


Start
Available for execution
Active
OpenMP worksharing loop
CU
Memory

Nymble-OMP


Start
Available for execution
Activated
Thread activated, context restored
No additional cycle for thread switch required
OpenMP worksharing loop
CU
Active
Stall, e.g. cache miss
Stalled
Thread deactivated
Context stored
Memory

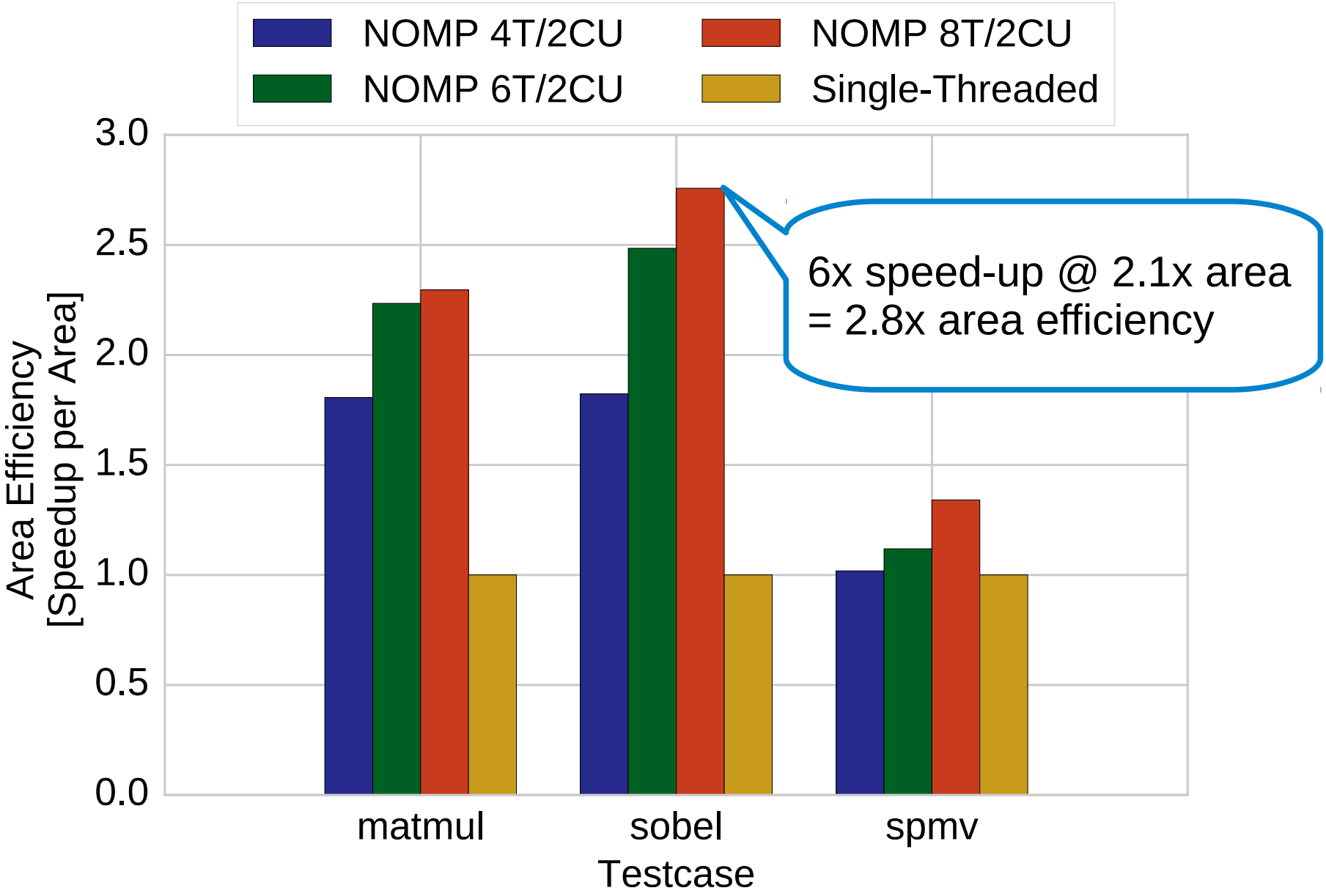
Nymble-OMP


Start
Available for execution
Activated
Active
Stall resolved
Stall, e.g. cache miss
Stalled
Memory request continues in background
• Thread deactivated
• Context stored
• Thread activated, context restored
• No additional cycle for thread switch required
OpenMP worksharing loop
CU
Memory

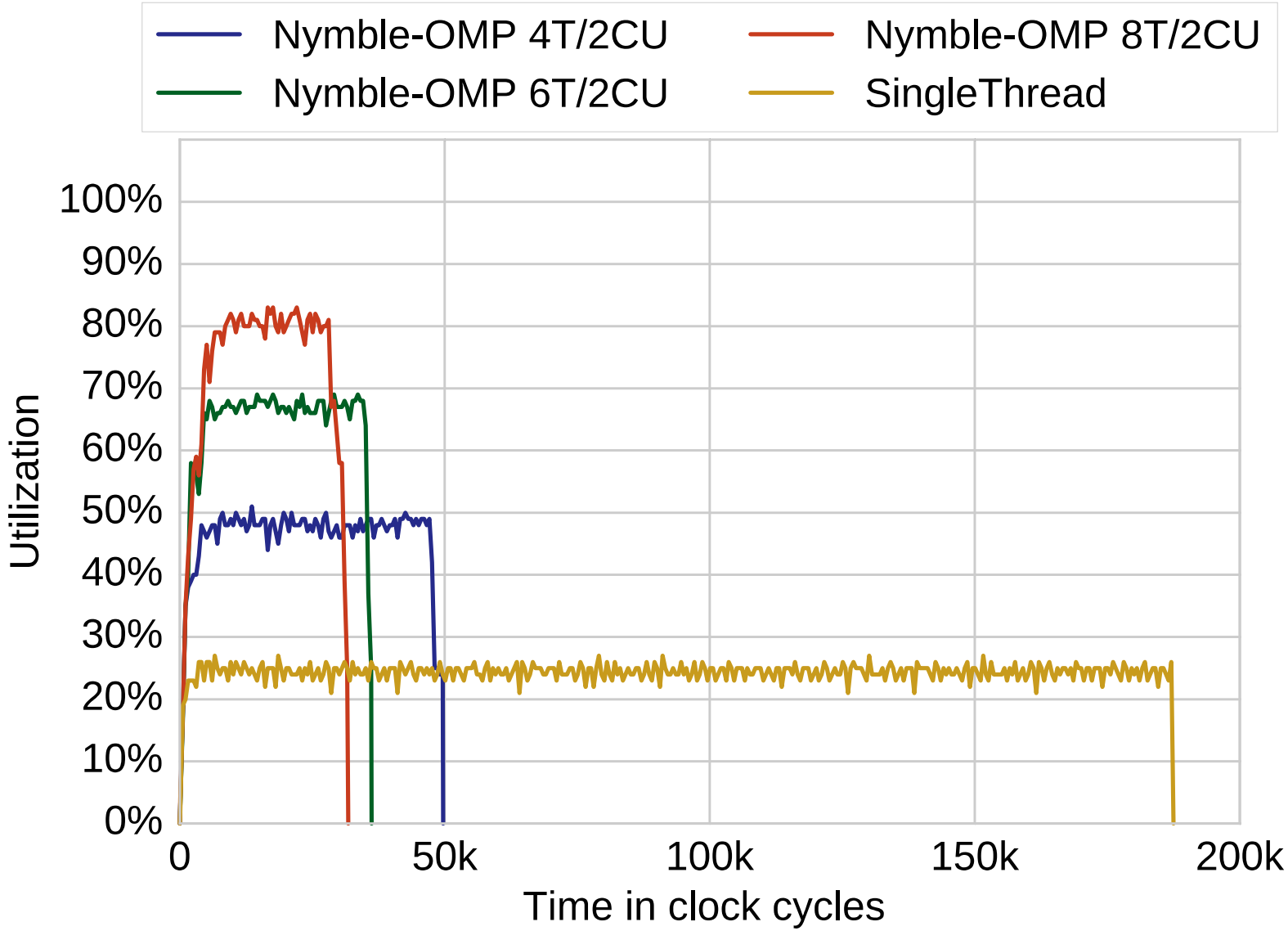
Evaluation

- Evaluation with OpenMP-parallelized matrix algorithms
- Synthesis with 2 CUs, 8 threads (total) for Virtex 7 (XC7VX690T):
  - Clock frequency: 100 MHz
  - Overall FPGA occupancy: 20-25%
  - ~5% overhead of multi-threading compared to single-threaded CU for:
    - Thread Switching Logic
    - Thread Context Storage

Evaluation


NOMP 4T/2CU
NOMP 8T/2CU
NOMP 6T/2CU
Single-Threaded
Area Efficiency [Speedup per Area]
3.0
2.5
2.0
1.5
1.0
0.5
0.0
6x speed-up @ 2.1x area
= 2.8x area efficiency
matmul
sobel
spmv
Testcase

Evaluation

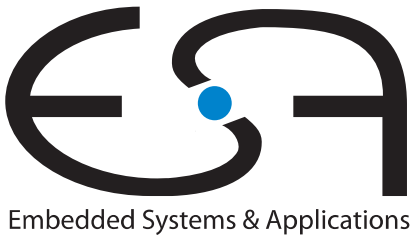


Example: sobel

Thank you!

Lukas Sommer

sommer@esa.tu-darmstadt.de



TECHNISCHE UNIVERSITÄT DARMSTADT